

AMD
Smarter Choice

AMD GPU Tools for games development

Holger Gruen
European Developer Relations
AMD Graphics Products Group
holger.gruen@amd.com

Material for many slides was provided by J. Zarge and S. Sowerby from the AMD graphics – Performance Tools - group

Develop conference, 24-26 July 2007

Overview

AMD Smarter Choice

AMD CPU Tools – a quick reminder

- CodeAnalyst, APL, ACML, ...

AMD GPU Tools

- GPU PerfStudio, GPU ShaderAnalyzer
- Tootle
- AMD content creation tools

GPU Tools demo

- DX9 case study
- DX10 demo on DXSDK samples

AMD
Smarter Choice

CPU Tools

CPU Tools: Overview

AMD Smarter Choice

AMD CodeAnalyst Performance Profiler

- Hotspot detection (various sampling methods)
- Call stack sampling
- Thread profiling, Pipeline simulation, etc. …
- Plug-in for Microsoft Visual Studio 05

Performance Libraries

- APL (AMD performance library)
- ACML (AMD core math library)

AMD
Smarter Choice

GPU PerfStudio

GPU PerfStudio: Overview

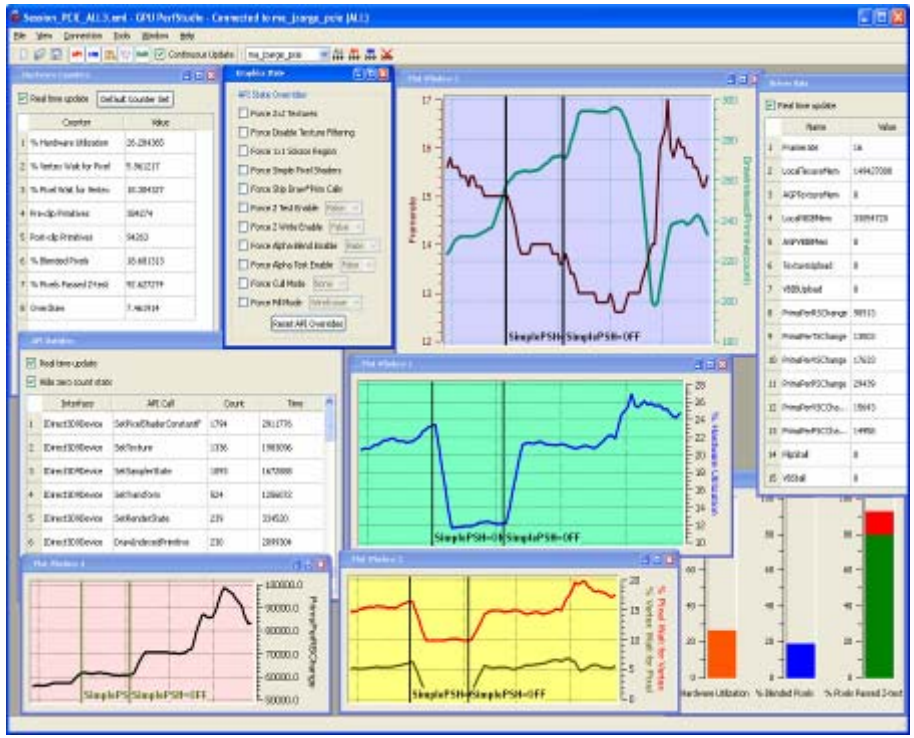
AMD Smarter Choice

GPU PerfStudio

Monitors GPU in real-time

- API statistics
- Hardware/driver data

of D3D apps



Visualize data in real-time

- in plots
- in bar charts

Client/Server architecture

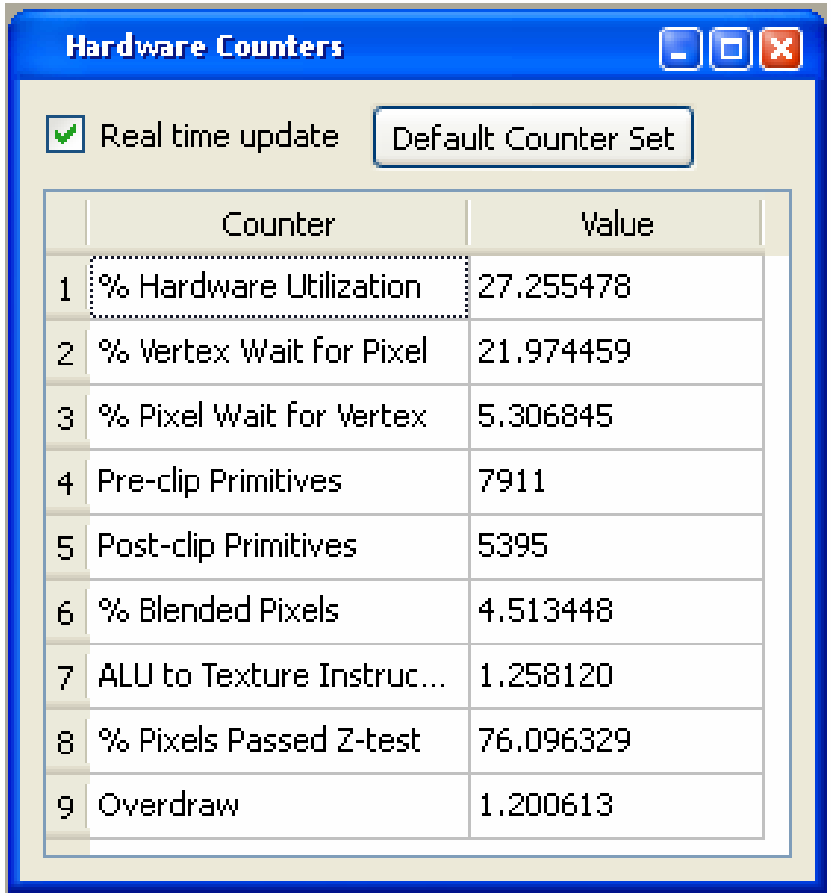
- Local or remote apps
- Remote app launching

Override rendering states in real-time

GPU PerfStudio Features: Hardware Counters

AMD Smarter Choice

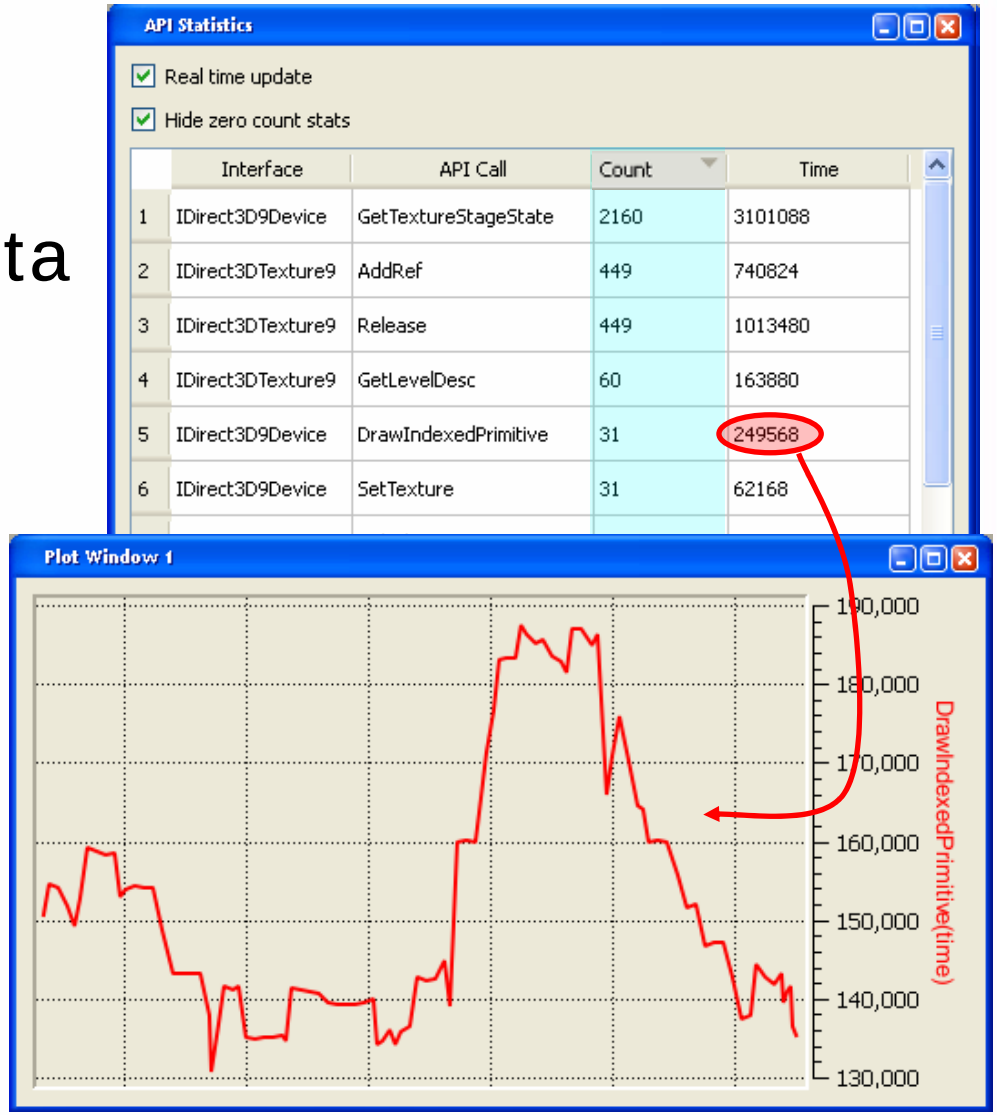
- % Hardware Utilization
- % Pixels passed Z-test
- Primitive counts
- Pixel vs. vertex bound
 - Still makes sense under DX10
- etc.



GPU PerfStudio Features: API Statistics

AMD Smarter Choice

- Per-frame API call data
 - D3D
 - OpenGL
- Sorting of API call by
 - call count
 - call timing



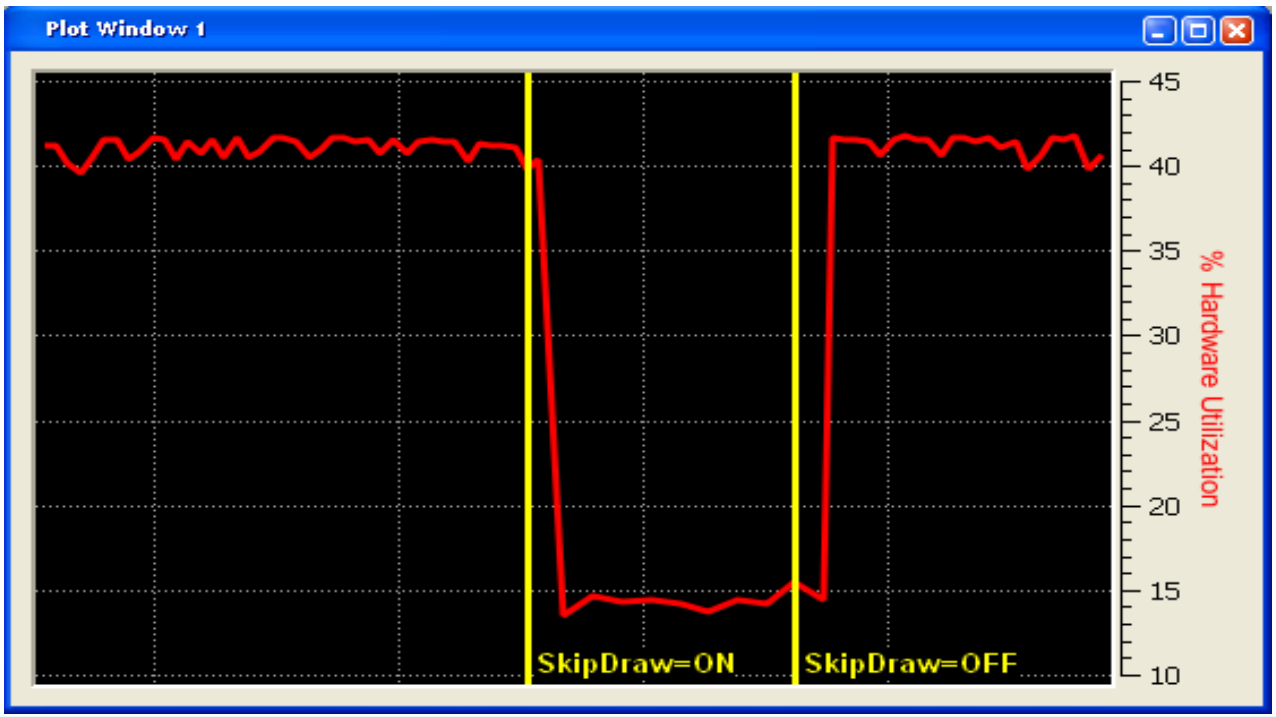
GPU PerfStudio Features: Data Plotting

AMD Smarter Choice

Marker lines for API state changes

Multiple data series on single plot

Plot properties customizable



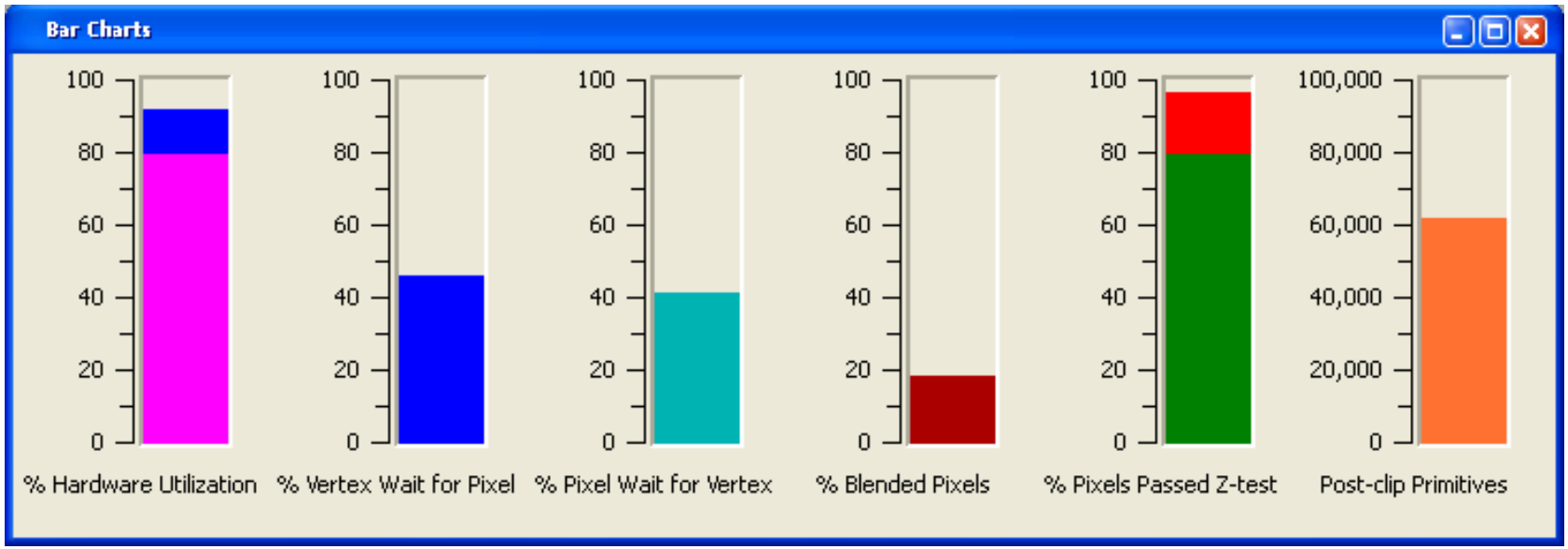
GPU PerfStudio Features: Bar Charts

AMD Smarter Choice

Customizable “alarm” level

Minimum and maximum range values

Data filtering



GPU PerfStudio Features: State Overrides

AMD Smarter Choice

Override	Description/Possible Bottleneck
Force 2x2 Textures	Is texture bandwidth (large textures) affecting performance?
Force Disable Texture Filtering	Are expensive texture filtering modes affecting performance?
Force 1x1 Scissor Region	Identifies vertex processing bottlenecks (by removing most pixel processing)
Force Simple Pixel Shaders	Identifies expensive pixel shaders
Force Skip Draw*Prim Calls	Identifies non-GPU bottlenecks (by removing most 3D graphics work)
Force Z Test Enable	Identifies z-order performance issues
Force Z Write Enable	Identifies z-order performance issues
Force Alpha Blend Enable	Identifies alpha-blending performance issues
Force Alpha Test Enable	Can identify problems related to early Z test
Force Cull Mode	Can show culling efficiency
Force Fill Mode	Used for debugging and identifying vertex density

GPU PerfStudio: Upcoming Features

AMD Smarter Choice

DX10/Vista/HD2000 support

- ability to analyze cutting edge applications on the newest hardware

Render target and non-RT state overrides

- Especially useful for image-space effects (depth of field, glows, render to texture)

Data filters – average median, min, max , derivative

- For gaining additional insight into your data

Plot lines for user defined markers

- Demonstrate the effect of state override on performance data

GPU PerfStudio: Upcoming Features (cont.)

AMD Smarter Choice

- Flexible bar charts
 - Control colors, sizes, alarm, range
- Flexible table cell formatting
 - Control cell size, color, font
- Application remembers settings
 - recent sessions, recent server machines, recent apps, etc.
- Selectable anti-aliasing
 - Fine tune the look and performance of GPU PerfStudio

AMD
Smarter Choice

GPU ShaderAnalyzer

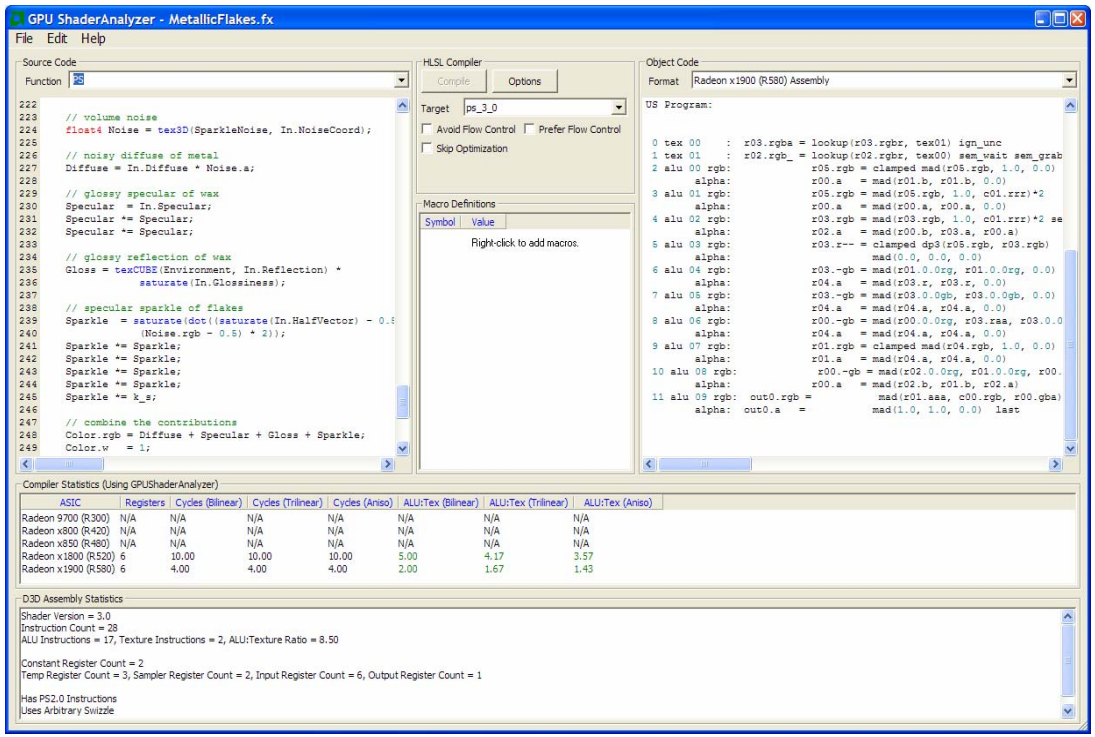
GPU ShaderAnalyzer: Feature Overview

AMD Smarter Choice

Shader performance analysis tool

Shader tuning environment

Instant perf. feedback as you tune your shaders



GSA Features: Performance Analysis

AMD Smarter Choice

Predicts shader perf. on range of AMD GPUs.

Analyzes compiled hardware instruction stream

- Analysis is tied to specific Catalyst driver releases.

Displays estimated cycle count & ALU:Texture ratio

Color codes ALU:Tex ratio for shader for

- ALU bound, Texture bound, Interpolator bound

Compiler Statistics (Using Catalyst 7.1)

ASIC	Registers	Cycles (Bilinear)	Cycles (Trilinear)	Cycles (Aniso)	ALU:Tex (Bilinear)	ALU:Tex (Trilinear)	ALU:Tex (Aniso)
Radeon 9700 (R300)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Radeon x800 (R420)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Radeon x850 (R480)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Radeon x1800 (R520)	4	8.00	8.00	8.00	2.00	1.67	1.43
Radeon x1900 (R580)	4	4.00	4.80	5.60	1.00	0.83	0.71

GSA Features: Performance Analysis (cont.)

AMD Smarter Choice

Considers static & dynamic flow control.

Considers the cost of each side of a branch

Calculate the minimum, maximum & average cycles

Factors in the expected flow control coherence

Currently only average cycles are displayed in GUI

- but min & max from command line and selectable in new versions

Estimates cost of texturing for bi-, tri-linear & aniso

- Cost based on typical texture fetch cost, not theoretical maximum

GSA Features: Hardware Disassembly

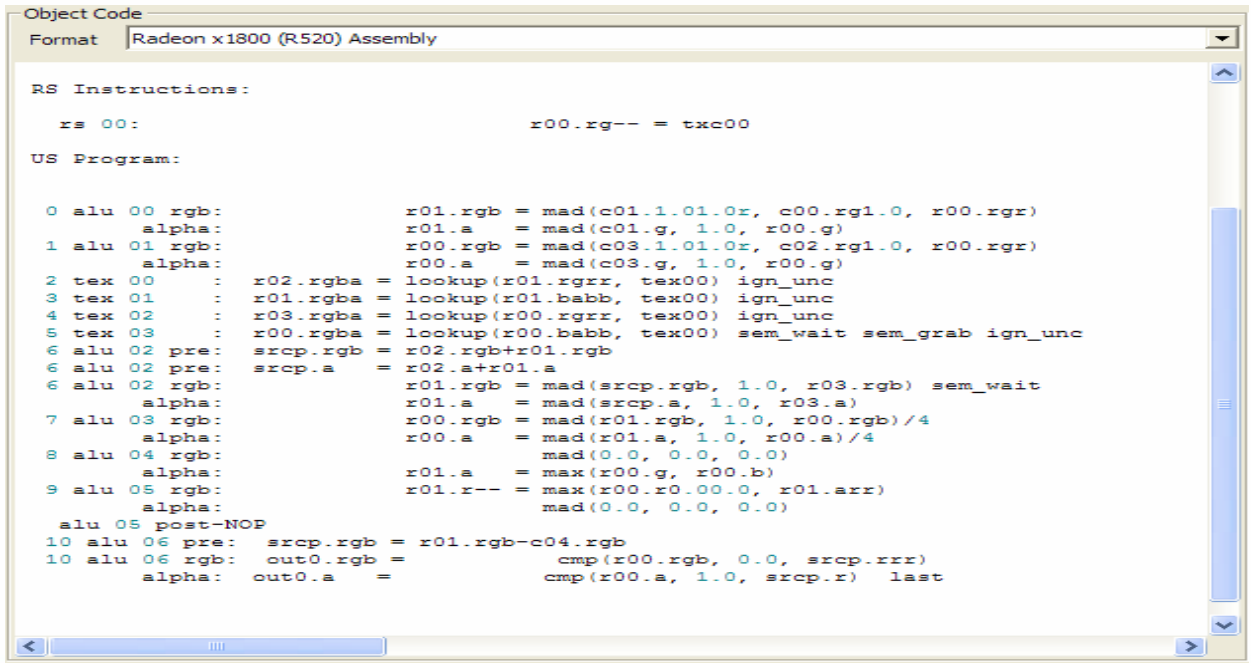
AMD Smarter Choice

View the actual shader as executed by the hardware

Shows the hardware shader optimized by the SC

See where your shader performance is going

Can also display D3D shader disassembly



GSA Features: Supported Shader Formats

AMD Smarter Choice

DX9 Pixel\Vertex Shaders

SM 1.1 – SM 3.0 Assembly Shaders

SM 2.0 – SM 3.0 HLSL Shaders

DX10 Pixel\Vertex\Geometry Shaders

GLSL Pixel\Vertex Shaders

arb_fp\arb_vp programs

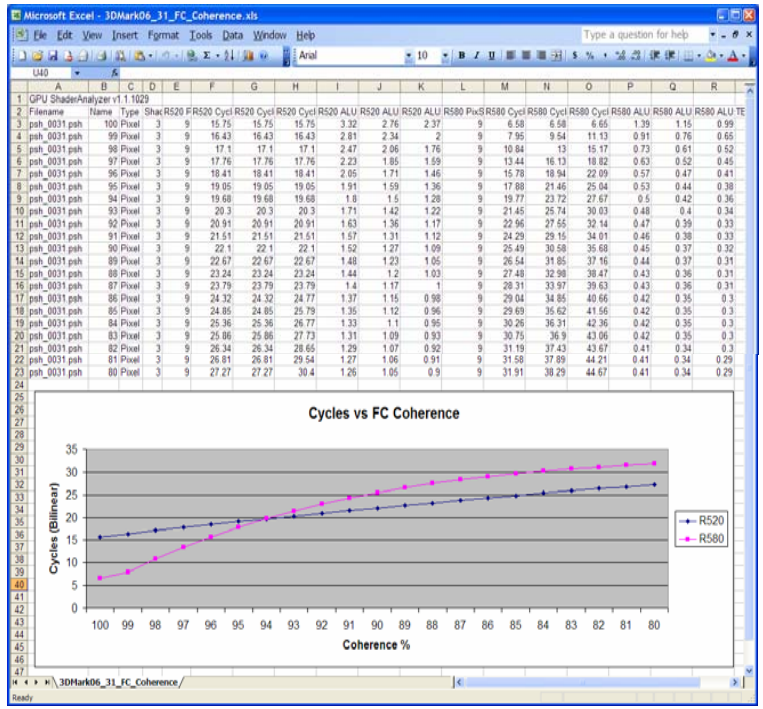
GSA Features: Other features

Options dialog to configure

- HLSL compiler to use
- ATI Shader Compiler Version
- GPUs to analyze performance for
- Options that control code analysis

Command line support

- performance analysis & hardware disassembly also available from command line – as mentioned earlier
- Analyze a single shader or a directory tree
- Output analysis to .csv file for further analysis within MS Excel



AMD
Smarter Choice

Tootle

Tootle: Overview

AMD Smarter Choice

A Triangle Order Optimization Tool

- Improves vertex-cache hit rate
- Reduces overdraw
- View independent

Library to integrate into your tool-chain

Simple to use and free

Tootle: Background

Based on I3D 06 paper by Nehab\Barczak\Sander

Uses D3DXOptimizeMesh for vtx cache optim.

Uses D3D for overdraw measurement

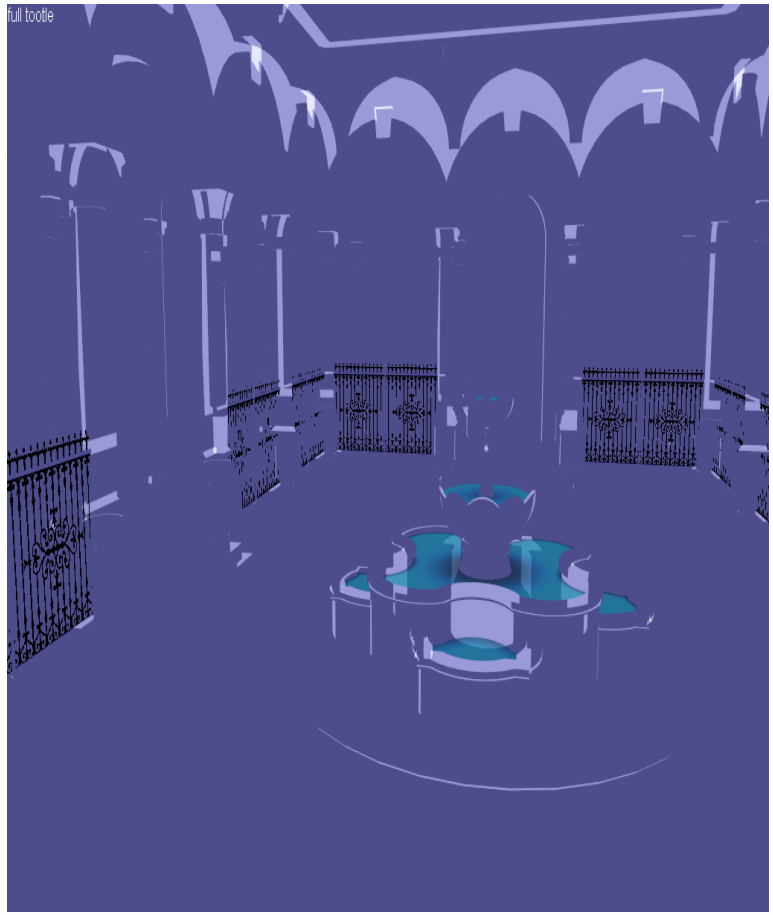
Example Scene: 70k polygons, 10 materials

- Reduced overdraw by factor of two.
- 3-7% performance increase compared to D3DXOptimizeMesh

Tootle: Overdraw Reduction







AMD
Smarter Choice

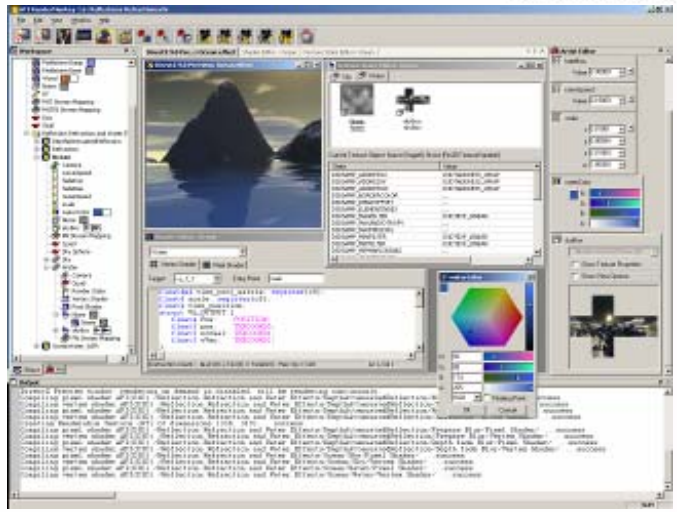
AMD Content Creation Tools

AMD Content Creation Tools

AMD Smarter Choice

RenderMonkey

- Shader development environment
- Supports HLSL, D3D asm and GLSL



The Compressonator

- Tool for compressing textures
- Creates mip-map levels
- DX10 supported

CubeMapGen

- Creates filtered seamless cube maps
- Uses angular extent filtering

NormalMapper

- Automatic normal map generation tool



GPU Tools demo: DX9 case study

Case Study: Settlers 6 (working title)

AMD Smarter Choice

Developer is BlueByte

- Development started spring ´05
- Will be released this summer

Empire building RTS game

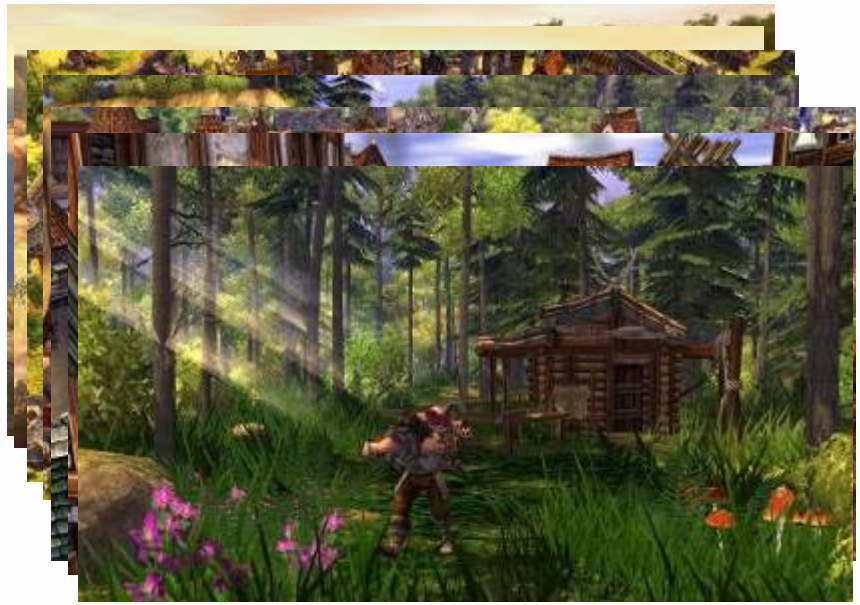
- Set in a medieval world
- Has a focus on city building
- Warfare part of the game

Single & Multi player modes

- Various multi player modes
- Story based and free-play maps

Complex graphics engine

- Highly dynamic world
- Lots of foliage



The Settlers®
©2006 Ubisoft Entertainment. All Rights Reserved. The Settlers, Blue Byte and the Blue Byte logo are trademarks of Red Storm Entertainment in the US and/or other countries. Ubisoft and the Ubisoft logo are trademarks of Ubisoft Entertainment in the US and/or other countries. Red Storm Entertainment Inc is a Ubisoft Entertainment company. Developed by Blue Byte Software.

Why were the tools used?

AMD
Smarter Choice

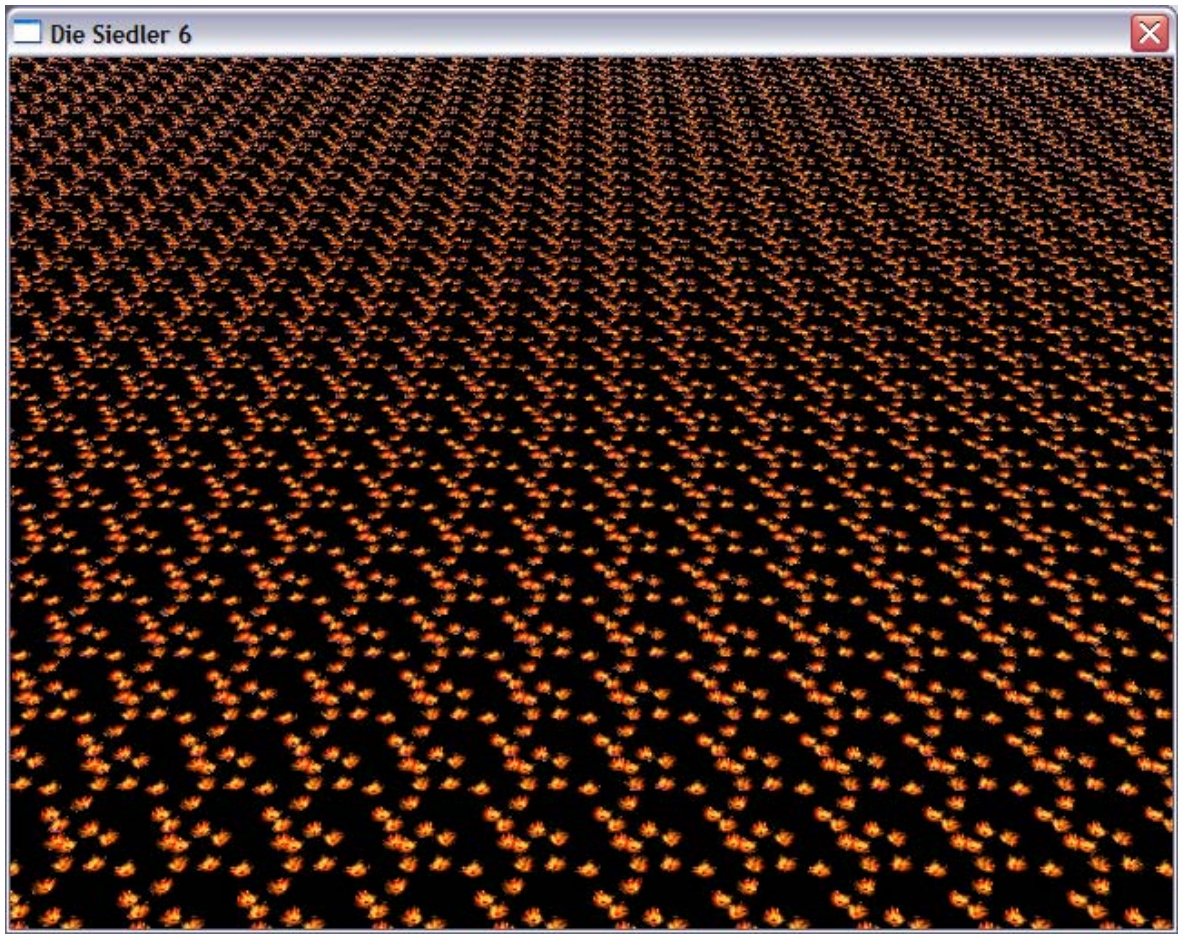
The Tools are not invasive!

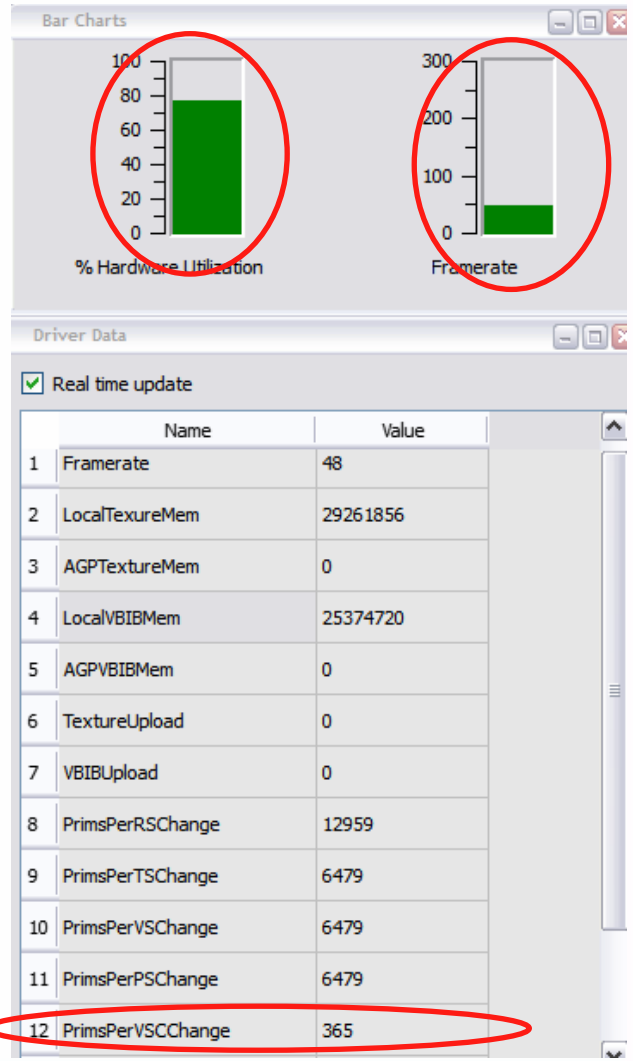
- No changes to your source code
- No additional risk even late during a project
- No additional Q&A ☺
- The tools are free!!

Tools would allow most efficient way to

- Assess impact of rendering techniques for new scene elements
- Only one scene element covered by this study

GPU PerfStudio: Rendering flowers - no instancing

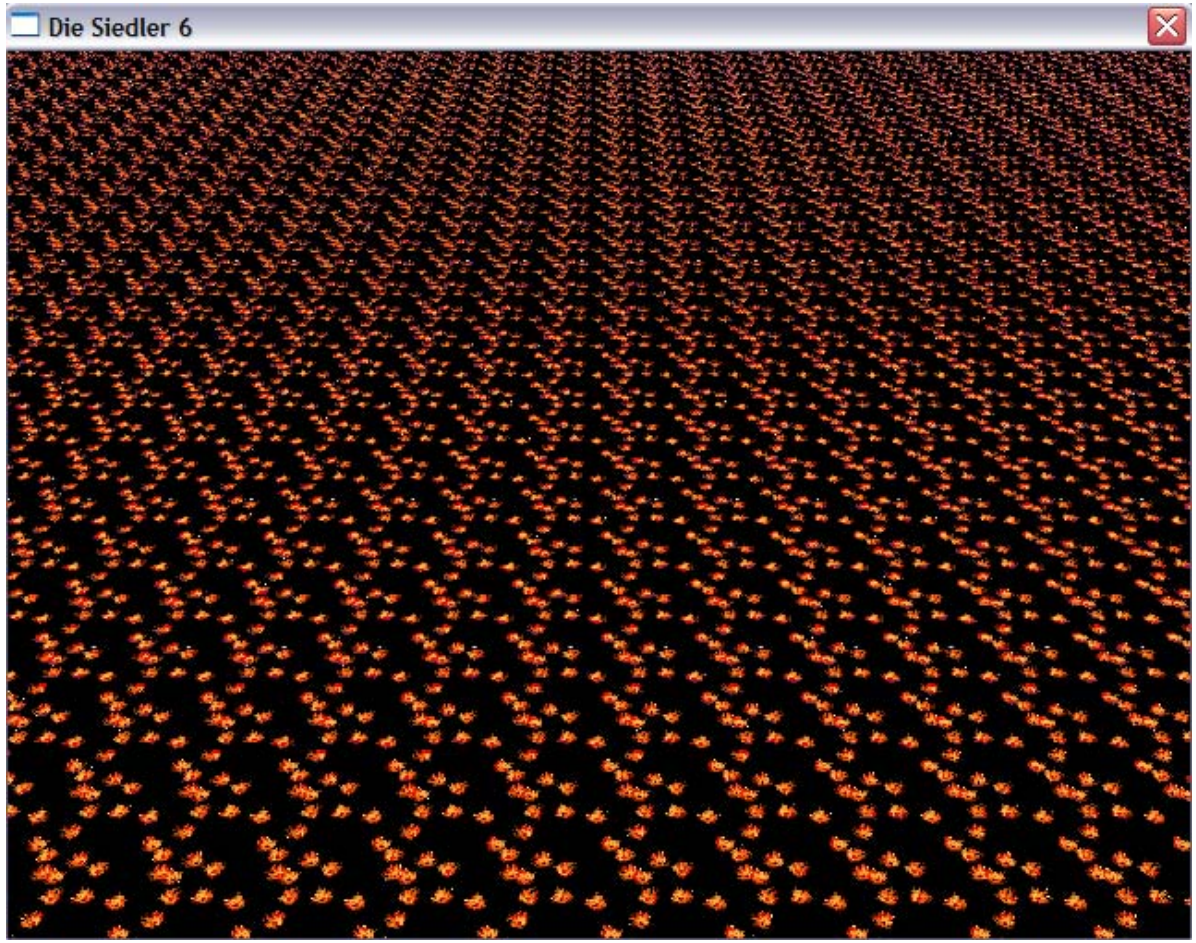


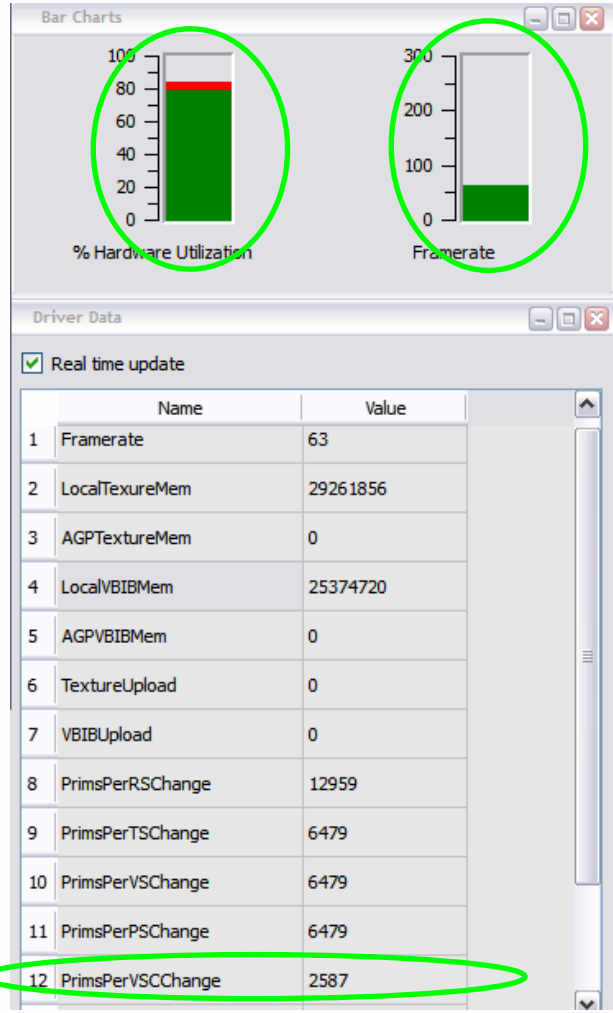


	Name	Value
1	Framerate	48
2	LocalTexureMem	29261856
3	AGPTextureMem	0
4	LocalVBIBMem	25374720
5	AGPVBIBMem	0
6	TextureUpload	0
7	VBIBUpload	0
8	PrimsPerRSChange	12959
9	PrimsPerTSChange	6479
10	PrimsPerVSChange	6479
11	PrimsPerPSChange	6479
12	PrimsPerVSCChange	365

GPU PerfStudio: Rendering flowers - instancing

AMD Smarter Choice

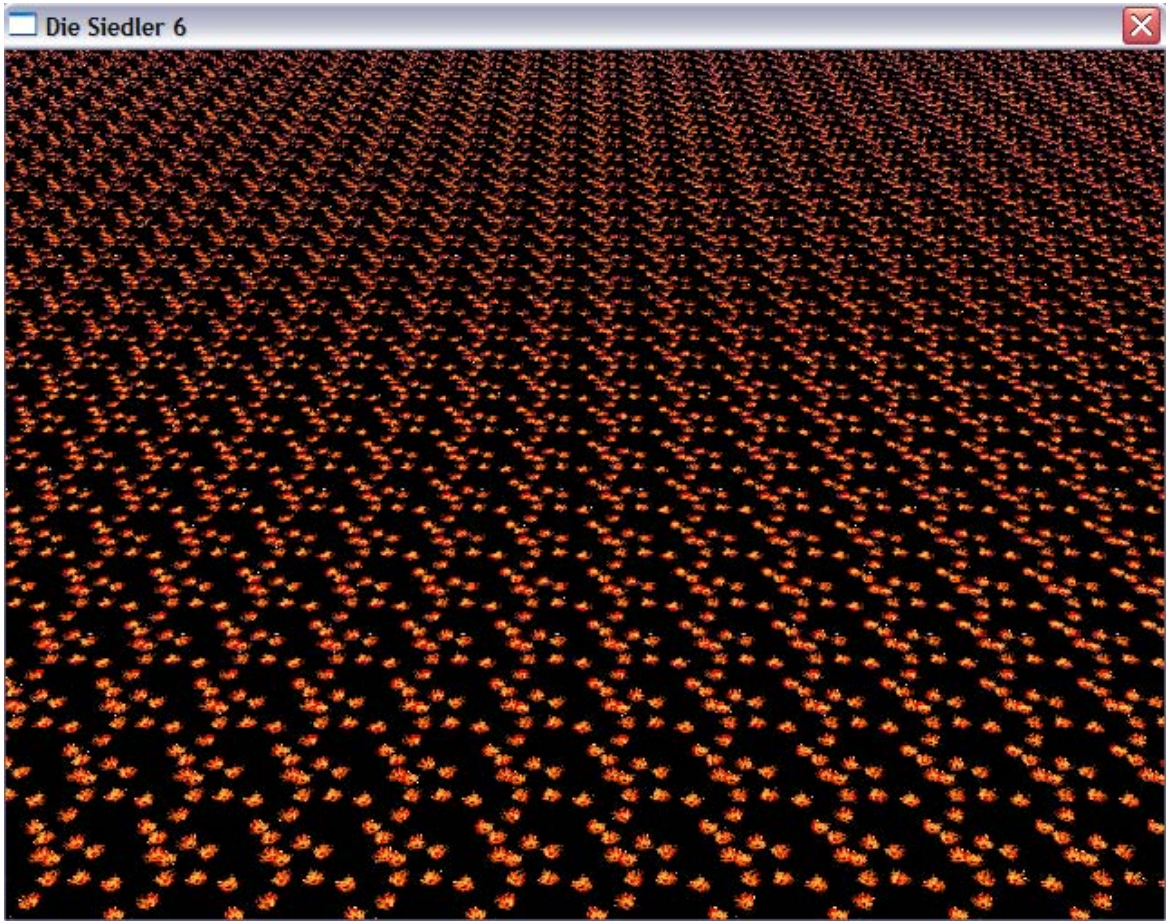




	Name	Value
1	Framerate	63
2	LocalTexureMem	29261856
3	AGPTextureMem	0
4	LocalVBIBMem	25374720
5	AGPVBIBMem	0
6	TextureUpload	0
7	VBIBUpload	0
8	PrimsPerRSChange	12959
9	PrimsPerTSChange	6479
10	PrimsPerVSChange	6479
11	PrimsPerPSChange	6479
12	PrimsPerVSCChange	2587

GPU PerfStudio: Rendering flowers – vertex limited

AMD Smarter Choice



Hardware Counters

Real time update | Default Counter Set

	Counter	Value
1	% Hardware Utili...	90.912636
2	% Vertex Wait f...	5.021636
3	% Pixel Wait for ...	76.222565
4	Pre-clip Primitives	108564
5	Post-clip Primitives	38835
6	% Blended Pixels	42.075352
7	% Pixels Passed ...	100.000000
8	Overdraw	0.583057

An experiment

AMD Smarter Choice

GPU ShaderAnalyzer - OrnamentalItem.fx

Lot's of pixels are waiting for vertices

- Reasons are complex scattering computations in vertex program
- Could pixels be more expensive at certain low resolutions?

GSA tells us PS could abide for more ALU on x1900

- Especially for Trilinear and Aniso
- There is a chance we can improve speed

Let's move scattering computations from VS to PS

- This is just an experiment!
- Won't go into details about the shader changes!

Compiler Statistics (Using Catalyst 7.1)

ASIC	Registers	Cycles (Bilinear)	Cycles (Trilinear)	Cycles (Aniso)	ALU:Tex (Bilinear)	ALU:Tex (Trilinear)	ALU:Tex (Aniso)	SCDev Registers	SVDev Insts
Radeon x850 (R480)	8	26.00	26.00	26.00	5.20	4.33	3.71	N/A	N/A
Radeon x1800 (R520)	7	18.00	18.00	18.00	3.60	3.00	2.57	N/A	N/A
Radeon x1900 (R580)	7	7.00	7.00	7.00	1.40	1.17	1.00	N/A	N/A

D3D Assembly Statistics

```
Shader Version = 3.0
Instruction Count = 120
ALU Instructions = 81, Texture Instructions = 15, ALU:Texture Ratio = 5.40

Constant Register Count = 4569
Temp Register Count = 20, Sampler Register Count = 10, Input Register Count = 9, Output Register Count = 3

Requires PS3.0
```

Reassessing the situation

GPU ShaderAnalyzer - OrnamentalItemPSScatter.fx

Ratios went up for X1900

- It is what we would expect ☺
- We are now ALU limited
- Good for future GPUs ☺

We are now green even for Aniso

Let's check what changed

- Let's run GPU PerfStudio again …

Compiler Statistics (Using Catalyst 7.1)

ASIC	Registers	Cycles (Bilinear)	Cycles (Trilinear)	Cycles (Aniso)	ALU:Tex (Bilinear)	ALU:Tex (Trilinear)	ALU:Tex (Aniso)	SCDev Registers	SVDev Insts
Radeon x850 (R480)	9	47.00	47.00	47.00	9.40	7.83	6.71	N/A	N/A
Radeon x1800 (R520)	8	31.00	31.00	31.00	6.20	5.17	4.43	N/A	N/A
Radeon x1900 (R580)	8	11.00	11.00	11.00	2.20	1.83	1.57	N/A	N/A

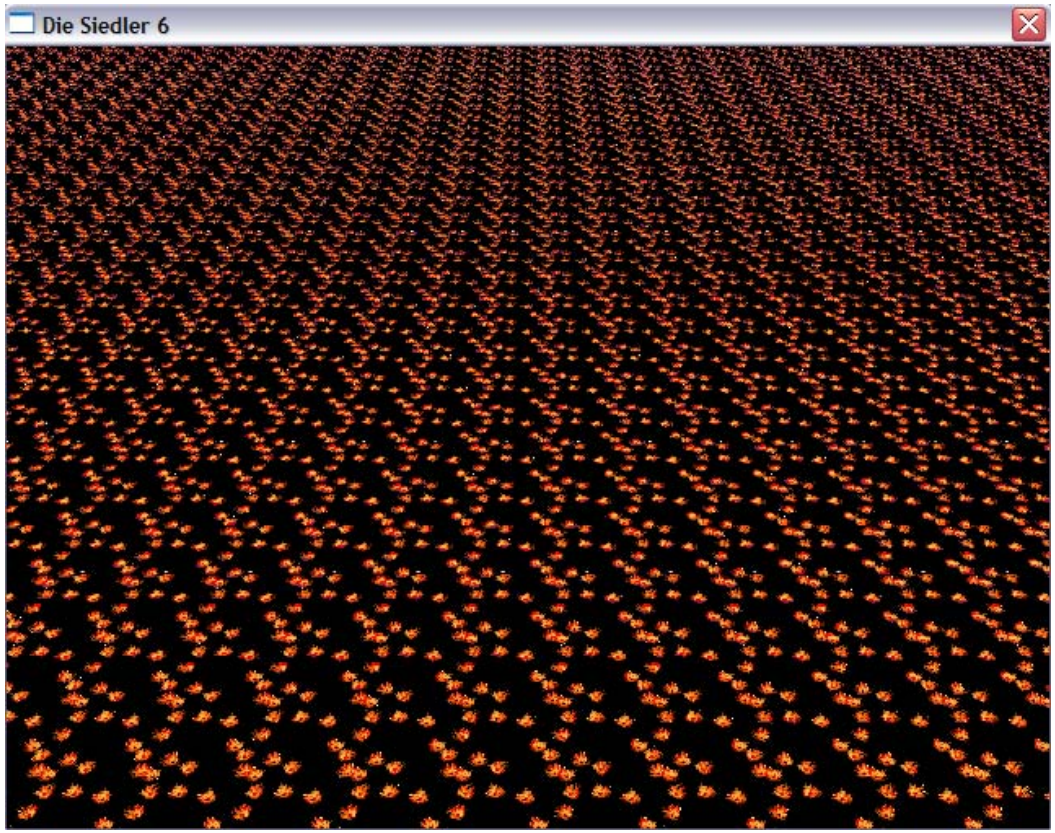
D3D Assembly Statistics

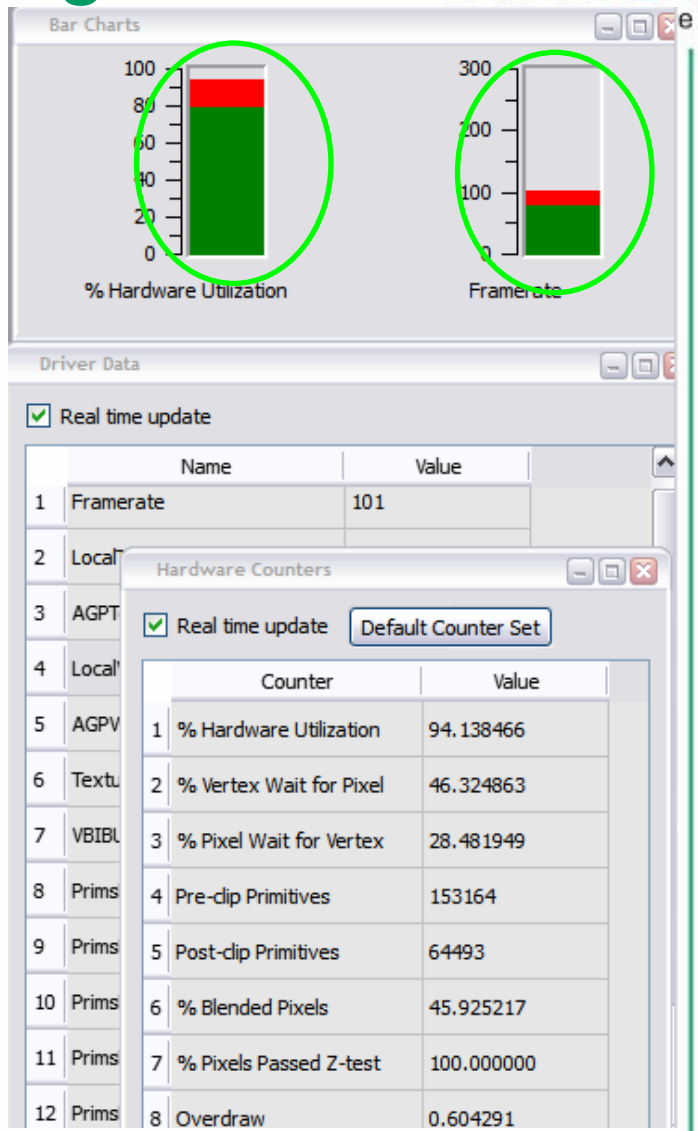
```
Shader Version = 3.0
Instruction Count = 195
ALU Instructions = 142, Texture Instructions = 20, ALU:Texture Ratio = 7.10

Constant Register Count = 4740
Temp Register Count = 20, Sampler Register Count = 16, Input Register Count = 13, Output Register Count = 5

Requires PS3.0
```

Let's run GPU PerfStudio again …

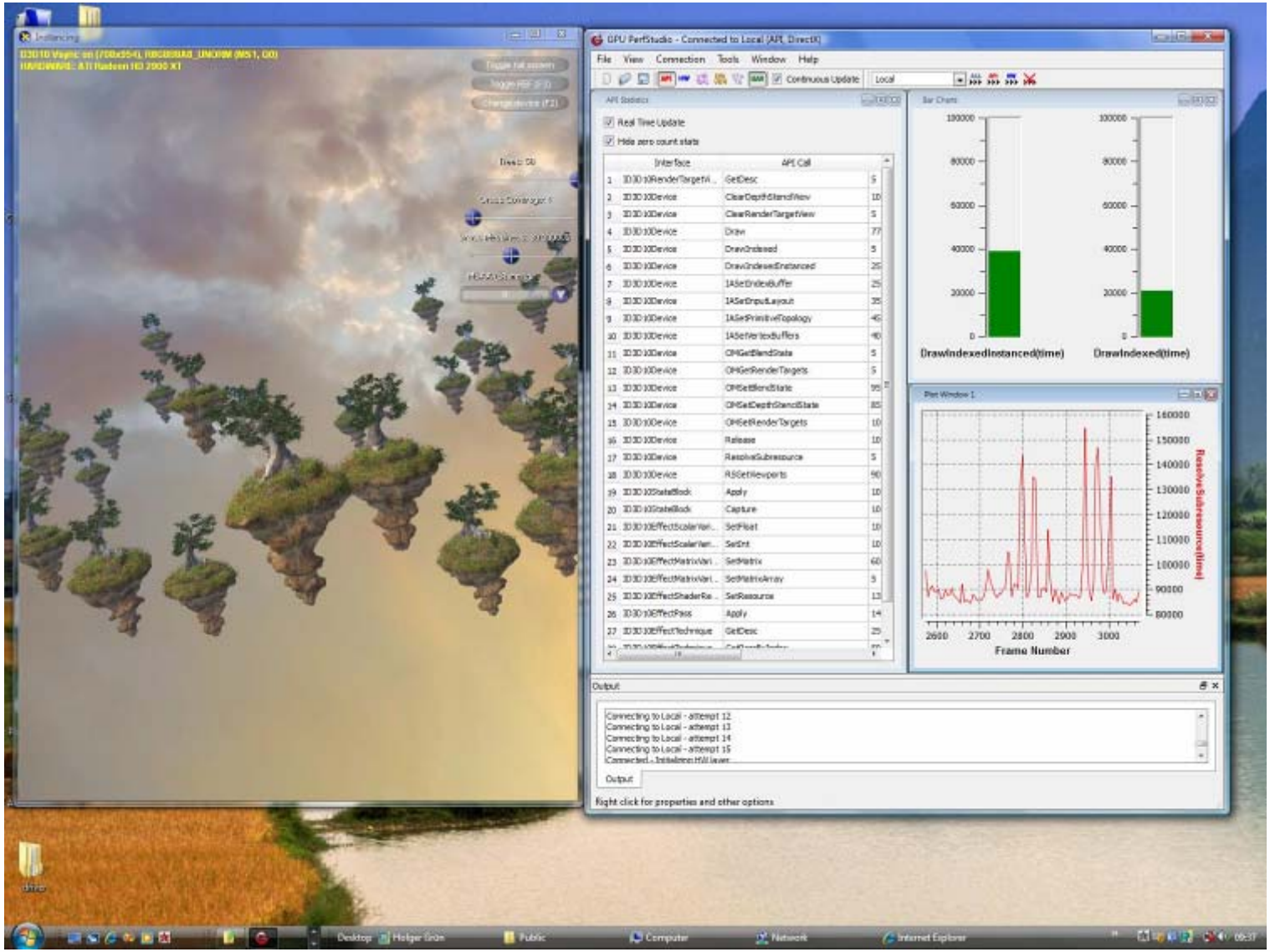




The experiment was a success …

GPU Tools demo: DX10 SDK samples

GPU PerfStudio on Vista / DX10



AMD
Smarter Choice

Questions?

Holger Gruen
European Developer Relations
AMD Graphics Products Group
holger.gruen@amd.com